Amendments to the Claims:

1. (Currently amended) A method of emulation of EEPROM memory in another non-volatile memory, the method comprising the steps:
after initiating the emulation, reserving two sectors of the non-volatile memory, serving ~~the~~ a function of ~~the~~ a current sector and ~~the~~ an auxiliary sector, creating ~~two buffers~~ a first buffer and a second buffer in ~~the~~ an operational memory, the first ~~of which stores always the~~ buffer for storing a most current image of the emulated EEPROM memory, and the second ~~stores the~~ buffer for storing a last patch, wherein the current sector of non-volatile memory ~~is organized in such a way that~~ has a part of the current sector ~~contains~~ containing ~~the~~ an original image of the emulated EEPROM memory, and a remaining part is successively filled in with ~~the~~ patches, the patches describing changes in ~~the~~ content of the original image of the emulated EEPROM memory, and wherein at ~~the~~ time, when a new patch cannot be appended to the current sector, the ~~functions~~ function of the ~~sectors~~ current sector and the auxiliary sector of non-volatile memory are changed, ~~thanks to which~~ the previously auxiliary sector of non-volatile memory is activated by saving the current image of the emulated EEPROM memory from the operational memory to a newly activated sector as a new original image of the emulated EEPROM memory, whereas, after a correct writing, ~~the~~ content of the previously current sector of non-volatile memory is erased.

2. (Currently amended) The method of emulation of the EEPROM memory according to claim 1, further comprising the step:
creating ~~the~~ a third buffer in the operational memory, which is used for storing ~~the patch after compression~~ a compressed version of the last patch.

3. (Currently amended) The method of emulation of the EEPROM memory according to claim 2, wherein at initiating of the emulation the current sector is selected, ~~next the~~ content of the auxiliary sector is erased and ~~next~~ the original image of the emulated EEPROM memory is fetched from the current sector of the non-volatile memory to the operational

memory, next, ~~the~~ a first patch is fetched from the current sector of non-volatile memory and its validity is checked, while if the first patch was not invalidated it is checked if the first patch data are compressed and in case when they are uncompressed, they are saved in the buffer of operational memory, which stores the most current image of the emulated EEPROM memory, ~~however, in opposite case,~~ whereas when the first patch data are compressed, before saving to the memory buffer, the first patch ~~is~~ data are decompressed, while in case, when the first patch ~~is~~ data are invalidated it is skipped, while ~~in the last point of the procedure~~ at an end of initiating emulation it is checked if there are still data for reading in the non-volatile memory and if there are ~~such~~ data for reading, ~~the~~ next patches are processed according to ~~the~~ a described algorithm.

4. (Currently amended) The method of emulation of the EEPROM memory according to claim 3, wherein the current sector and the auxiliary sector are selected by setting the first sector as the current ~~one~~ sector, next it is checked if ~~the~~ saved data have a correct format and if the saved data were correctly saved, and if ~~in case of such check~~ it appears that there are incorrect data in the first sector, the second sector is set as the current sector, while the first sector is set as the auxiliary sector, ~~however, in the opposite case, that is when there~~ whereas when are correct data in the first sector, the second sector is set as the current sector, and next it is checked if the saved data have a correct format and if the data were saved correctly, and if ~~as~~ a result of ~~such~~ check ~~it appears~~ shows that there are incorrect data in the second sector, the first sector is set as the current sector, while the second sector is set as the auxiliary sector, ~~while, in the opposite case~~ whereas if a result of check shows that there are incorrect data in the second sector one of the first sector and the second sector, in which there is more free space, is set as the current sector.

5. (Currently amended) The method of emulation of the EEPROM memory according to claim 2, ~~the~~ a process of writing a new patch starts from preparing data for updating the content of the non-volatile memory, while the data are also saved in the buffer of the operational memory, and next it is checked if the size of the new patch is greater than ~~the~~ a

set value and if it is greater, the new patch is compressed, then it is checked if the result of compression corresponds to the required assumptions of reducing the new patch size, whereas in case of compliance the compressed patch is further processed, and in the opposite case the uncompressed patch is processed, next, it is checked if in the current sector of non-volatile memory there is sufficient space for saving the new patch and in case of sufficient space present, the new patch is saved, and in case of a lack of sufficient free space the current sector is changed into the ~~second one~~ auxiliary sector whereas the auxiliary sector is changed to the current sector and the new original image of the emulated EEPROM memory is saved in the ~~second~~ current sector of the non-volatile memory, while the content of the ~~second~~ auxiliary sector in the Flash memory is erased.

6. (Currently amended) The method of emulation of the EEPROM memory according to claim 5, wherein in the process of preparing the new patch of the content of the non-volatile memory a preparation of the new patch for saving in the memory is made, next the data are saved in the operational memory, which stores the most current image of the EEPROM memory, at the same time the prepared patch is stored in the buffer of the operational memory, and next it is checked whether the patch, which was last saved in the non-volatile memory is valid while, if the patch is invalidated the writing process the new patch is continued, while in case when the previously saved patch is valid, it is checked if the currently processed patch reverses the changes introduced by the saving of the previous patch, whereas, if the processed patch did not reverse these changes, the writing process of the new patch is continued, and, in the opposite case, when the patch reverses the changes introduced by the saving of the previous patch, the value of the bit, which invalidates previously saved patch, is changed and the saving of the new patch to the non-volatile memory is ~~cancelled~~ canceled.

7. (Currently amended) The method of emulation of the EEPROM memory according to claim 2, wherein ~~the~~ a format of the patch consists of four fields, the first of which is ~~the~~ a patch header, the second field appearing in case of patches containing many groups of data is

a field of ~~the~~ a size of data group, the third one appearing only in case of uncompressed patches is an offset field of data in relation to ~~the~~ an initial address, while the last field in the patch is a data field, while the patch contains many data groups, of which every ~~one~~ group is saved under a different memory address, and ~~the~~ values of the offset of patch data groups contain an offset in relation to the final address of the previous data group, of which only the first offset defines the absolute address, while the next values are relative addresses in relation to the previous data group.

8. (Currently amended) The method of emulation of the EEPROM memory according to claim 7, wherein the compressed patch does not contain the offset field, ~~while the~~ whereas value of the offset is read only after decompression of the patch.

9. (Currently amended) The method of emulation of the EEPROM memory according to claim 7, wherein the format of the header of the patch consist of ~~the~~ a start bit, changed at the time when preparation of the patch for saving is started, ~~the~~ a bit of correct writing of the size and format, changed after the size value and format of data are correctly recorded, the bit of correct writing, changed after the whole patch is correctly recorded in the non-volatile memory, ~~the~~ an invalidation bit, which is changed after the patch is invalidated and ~~the~~ a field defining the format of the patch and the field, defining the total amount of data in the patch, not considering the size of the header.

10. (Previously presented) The method of emulation of the EEPROM memory according to claim 9, wherein the format of the patch is defined as a single update, or as a multi data groups patch or as a compressed patch.

11. (Previously presented) The method of emulation of the EEPROM memory according to claim 5, wherein for the format of the header of the patch, containing the start bit, changed in the time of starting preparation of the patch for recording, the bit of correct writing of the size and format, changed after the data size or format are correctly recorded, the bit of correct

writing, changed after the whole patch is correctly saved in the non-volatile memory, the invalidation bit, changed after the patch is invalidated and the field, defining the format of the patch and the field, defining the quantity of data in the patch, not considering the size of the header, saving of the patch starts from a change of the value of the start bit, after which the size and the type of the patch is recorded and if an error occurs the procedure ends, while if the writing of the fields is correct, the value of the bit in the field of correct writing of the size and format is changed and separate data groups are further recorded, and if an error occurs the procedure ends, while, if the so-far-writing is correct the value of the bit of correct writing is changed and at this moment the procedure ends and the patch is correctly recorded in the non-volatile memory.

12. (Currently amended) A method of software emulation of EEPROM memory in another non-volatile memory of a device with an operational memory, the method comprising the steps of:
creating in ~~an~~ the operational memory a first buffer for storing a current image of an emulated EEPROM memory and a second buffer for storing ~~the~~ a last patch;
applying two sectors of non-volatile memory as a current sector and an auxiliary sector;
storing an image of the emulated EEPROM memory in a part of the current sector;
filling in the remaining part of the current sector with one or more patches describing changes in content of the image of the emulated EEPROM memory and storing the current image of the emulated EEPROM memory in the first buffer and repeating the steps of filling in and storing the current image until a new patch cannot be added to the current sector;
changing a sector being previously the auxiliary sector into the current sector and a sector being previously the current sector into the auxiliary sector;
saving the current image of the emulated memory from the first buffer of the operational memory to the current sector being newly activated as the image of the emulated memory;
erasing content of the auxiliary sector being newly activated after a correct saving of the current image of the emulated memory; and

returning to a step of the filling in the remaining part of the current sector with one or more patches describing changes in the content of the image of the emulated EEPROM memory.

13. (Currently amended) The method of emulation of the EEPROM memory according to claim 12, wherein the step of the applying the two sectors of non-volatile memory as the current sector and the auxiliary sector comprises the steps of:
selecting the current sector from the two sectors of the non-volatile memory, the current sector containing the image of the emulated EEPROM memory and the patches describing changes in the content of the image of the emulated EEPROM memory;
erasing content of the auxiliary sector;
fetching the image of the emulated EEPROM memory from the current sector of the non-volatile memory to the operational memory;
fetching a first patch of the patches from the current sector of non-volatile memory;
checking validity of the first patch;
saving data of the first patch in the first buffer of the operational memory when the data is not invalidated and uncompressed and saving the data after decompressing when the data is not invalidated and compressed and skipping the first patch when the first patch is invalidated;
checking if in the current sector of the non-volatile memory are another patches; and
processing remained patches in the current sector of the non-volatile memory like the first patch.

14. (Currently amended) The method of emulation of the EEPROM memory according to claim 13, wherein the step of the selecting the current sector from the two sectors of the non-volatile memory comprises the steps of:
setting initially a first sector of the two sectors of the non-volatile memory as the current sector;
checking if the data saved in the first sector ~~haves~~ have a correct format and were correctly saved;

setting initially a second sector of the two sectors of the non-volatile memory as the current sector when the data saved in the first sector ~~haves~~ have an incorrect format or were incorrectly saved;
checking if the data saved in the second sector ~~haves~~ have a correct format and were correctly saved;
setting the first sector of the two sectors of the non-volatile memory as the current sector and setting the second sector of the two sectors of the non-volatile memory as the auxiliary sector when the data saved in the second sector ~~haves~~ have an incorrect format or were incorrectly saved; and
setting one sector of the two sectors having more free space as the current sector when the data saved in the first sector and the second sector ~~haves~~ have an incorrect format or were incorrectly saved.

15. (Currently amended) The method of emulation of the EEPROM memory according to claim 12, wherein a writing process of the new patch comprises the steps of:
preparing data for updating the content of the non-volatile memory and saving the data in the first buffer of the operational memory;
checking size of the data;
compressing the data when the size of the data is greater than a set value;
checking free space for saving the data in the current sector of the non-volatile memory;
saving the data as the new patch when the free space in the current sector of the non-volatile memory has a sufficient size for saving the new patch;
changing the auxiliary sector into the current sector and a sector being previously the current sector into the auxiliary sector when in the current sector of the non-volatile memory is no sufficient free space for saving the new patch and saving the new patch in a sector being previously the auxiliary sector and erasing ~~a~~ content of the sector being previously the current sector.

16. (Previously presented) The method of emulation of the EEPROM memory according to claim 12, wherein a format of the patches consists of a first field being a patch header, a second field containing many groups of data and having a size of data groups, a third field containing an offset field of data in relation to an initial address and fourth field containing data, wherein each patch contains many data groups saved under different memory addresses and wherein values of offsets of the data groups contain an offset in relation to a last address of a previous data group, of which only the first offset defines an absolute address, whereas next values are relative addresses in relation to the previous data group.

17. (Currently amended) The method of emulation of the EEPROM memory according to claim 16, wherein the format of the patch header consist of a start bit changed at time when preparation of the patch for saving is started, a bit of correct writing of size and format changed after a size value and format of data are correctly recorded, a bit of correct writing changed after the whole patch is correctly recorded in the non-volatile memory, an invalidation bit changed after the patch is invalidated and ~~the~~ a field defining ~~the~~ a format of the patch and the field defining total amount of data in the patch, not considering the size of the header.

18. (Previously presented) A method of a software emulation of an EEPROM memory in another non-volatile memory of a device with an operational memory, the method comprising the steps of:
creating in the operational memory a first buffer for storing a current image of the emulated EEPROM memory;
setting two sectors of the non-volatile memory as a current sector and an auxiliary sector;
storing an image of the emulated EEPROM memory in a part of the current sector;
updating the emulated EEPROM memory by adding, to the remaining part of the current sector, one or more patches, describing changes in content of the current image of the emulated EEPROM memory, and storing the current image of the emulated EEPROM memory, including the changes made by the one or more patches, in the first buffer;

repeating the step of updating the emulated EEPROM memory until a new patch cannot be added to the current sector;
setting the auxiliary sector as the current sector and the current sector as the auxiliary sector;
storing the current image of the emulated memory from the first buffer of the operational memory to the current sector; and
after the storing of the current image, from the first buffer, in the current sector has been finished successfully, erasing the auxiliary sector.

19. (Previously presented) The method according to claim 18 further comprising the step of:
returning to the step of updating the emulated EEPROM memory after the step of erasing the auxiliary sector.

20. (Previously presented) The method according to claim 18 wherein a second buffer is created, in the operational memory, for storing the last patch describing changes in content of the current image of the emulated EEPROM memory.